

What Is Claimed Is:

- 1 1. A method for determining a class dependency that identifies a
2 supporting class on which a target class depends, wherein the target class is
3 defined in an object-oriented programming language, comprising:
4 receiving a representation of the target class;
5 creating a model of the target class from the representation;
6 analyzing the model to detect references to the supporting class;
7 if a supporting class is detected, determining a class dependency for the
8 supporting class; and
9 creating a list of dependent classes for the target class and supporting
10 classes.
- 1 2. The method of claim 1, further comprising, identifying classes that
2 an object depends upon by:
3 receiving a representation of the object;
4 serializing the referenced object;
5 parsing the data resulting from the object serialization to identify classes
6 referenced from the target object's properties, configuration, or state; and
7 determining the dependent classes of the referenced object.
- 1 3. The method of claim 1, further comprising saving the list of
2 dependent classes to a storage structure.
- 1 4. The method of claim 3, wherein the storage structure is one of a
2 hash table and a database.

1 5. The method of claim 1, wherein creating the list of dependent
2 classes includes creating one of a distribution list and a distribution file.

1 6. The method of claim 2, further comprising:
2 inserting the object into an object database;
3 determining if the target class and supporting classes for the target class
4 are in the class path; and
5 adding the target class and supporting classes for the target class to the
6 class path if necessary.

1 7. The method of claim 2, further comprising:
2 retrieving the object from the object database;
3 determining if the target class and supporting classes for the target class
4 are in the class path; and
5 adding the target class and supporting classes for the target class to the
6 class path if necessary.

1 8. The method of claim 1, further comprising filtering the list of
2 identified classes to remove duplicate and core class references.

1 9. The method of claim 1, further comprising saving the list of
2 dependent classes of the target class as well as the list of dependent classes of the
3 supporting classes in cache to facilitate subsequent lookups of dependent classes
4 of the target class.

1 10. A computer-readable storage medium storing instructions that
2 when executed by a computer cause the computer to perform a method for

3 determining a class dependency that identifies a supporting class on which a target
4 class depends, wherein the target class is defined in an object-oriented
5 programming language, the method comprising:
6 receiving a representation of the target class;
7 creating a model of the target class from the representation;
8 analyzing the model to detect references to the supporting class;
9 if a supporting class is detected, determining a class dependency for the
10 supporting class; and
11 creating a list of dependent classes for the target class and supporting
12 classes.

1 11. The computer-readable storage medium of claim 10, wherein the
2 method further comprises, identifying classes that an object depends upon by:
3 receiving a representation of the object;
4 serializing the referenced object;
5 parsing the data resulting from the object serialization to identify classes
6 referenced from the target object's properties, configuration, or state; and
7 if a target class is identified, determining the dependent classes of the
8 target class.

1 12. The computer-readable storage medium of claim 10, wherein the
2 method further comprises saving the list of dependent classes to a storage
3 structure.

1 13. The computer-readable storage medium of claim 12, wherein the
2 storage structure is one of a hash table and a database.

1 14. The computer-readable storage medium of claim 10, wherein
2 creating the list of dependent classes includes creating one of a distribution list
3 and a distribution file.

1 15. The computer-readable storage medium of claim 11, wherein the
2 method further comprises:
3 inserting the object into an object database;
4 determining if the target class and supporting classes for the target class
5 are in the class path; and
6 adding the target class and supporting classes for the target class to the
7 class path if necessary.

1 16. The computer-readable storage medium of claim 11, wherein the
2 method further comprises:
3 retrieving the object from the object database;
4 determining if the target class and supporting classes for the target class
5 are in the class path; and
6 adding the target class and supporting classes for the target class to the
7 class path if necessary.

1 17. The computer-readable storage medium of claim 10, wherein the
2 method further comprises filtering the list of identified classes to remove
3 duplicate and core class references.

1 18. The computer-readable storage medium of claim 10, wherein the
2 method further comprises saving the list of dependent classes of the target class as

3 well as the list of dependent classes of the supporting classes in cache to facilitate
4 subsequent lookups of dependent classes of the target class.

1 19. An apparatus that determines a class dependency that identifies a
2 supporting class on which a target class depends, wherein the target class is
3 defined in an object-oriented programming language, comprising:
4 a receiving mechanism that is configured to receive a representation of the
5 target class;
6 a modeling mechanism that is configured to create a model of the target
7 class from the representation;
8 an analysis mechanism that is configured to analyze the model to detect
9 references to the supporting class;
10 a supporting mechanism that is configured to determine a class
11 dependency for the supporting class; and
12 a listing mechanism that is configured to create a list of dependent classes
13 for the target class and supporting classes.

1 20. The apparatus of claim 19, wherein the receiving mechanism is
2 additionally configured to receive a representation of an object;
3 a serializing mechanism is configured to serialize the referenced object;
4 a parsing mechanism configured to parse the data resulting from the object
5 serialization to identify classes referenced from the target object's properties,
6 configuration, or state; and
7 a supporting mechanism that is configured to determine the dependent
8 classes of the target class.

1 21. The apparatus of claim 19, wherein the listing mechanism is
2 configured to save the list of dependent classes to a storage structure.

1 22. The apparatus of claim 21, wherein the storage structure is one of a
2 hash table and a database.

1 23. The apparatus of claim 19, wherein the listing mechanism is
2 configured to create the list of dependent classes, including creating one of a
3 distribution list and a distribution file.

1 24. The apparatus of claim 20, further comprising:
2 an insertion mechanism configured to insert the object into an object
3 database;
4 a determining mechanism configured to determine if the target class and
5 supporting classes for the target class are in the class path; and
6 an adding mechanism configured to add the target class and supporting
7 classes for the target class to the class path if necessary.

1 25. The apparatus of claim 20, further comprising:
2 a retrieving mechanism configured to retrieve the object from an object
3 database;
4 a determining mechanism configured to determine if the target class and
5 supporting classes for the target class are in the class path; and
6 an adding mechanism configured to add the target class and supporting
7 classes for the target class to the class path if necessary.

1 26. The apparatus of claim 19, further comprising a filtering
2 mechanism configured to filter the list of identified classes to remove duplicate
3 and core class references.

1 27. The apparatus of claim 19, further comprising a saving mechanism
2 configured to save the list of dependent classes of the target class as well as the
3 list of dependent classes of the supporting classes in cache to facilitate subsequent
4 lookups of dependent classes of the target class